

WHAT IS CLAIMED IS:

1. A communications interface between an optical network and a computer processor or a communication switch, said interface is adapted to synchronously receive and transmit a serial data stream, said communications interface comprising:
- 5 a first queue manager segment adapted to remove a first portion of the data from a frame of the received data stream and transfer the removed data to the computer processor;
 - 10 a second queue manager segment adapted to transfer data from the computer processor into a frame of the transmitted data stream;
 - 15 a pass-through buffer segment that transfers a second portion of the received frame to the transmitted frame, absent transferring the received frame to the computer processor;
 - 20 a first frame index adapted to compute a unique physical address within the received frame of a byte in the received data stream;
 - 25 a second frame index adapted to compute a unique physical address within the transmitted frame of a byte in the transmitted data stream;
 - 30 first pointer control logic adapted to maintain a pointer to a payload portion of the received frame;
 - second pointer control logic adapted to maintain a pointer to a payload portion of the transmitted frame;
 - a first index adapted to compute a unique logical address for a byte within the payload portion of the received frame; and
 - a second index adapted to compute a unique logical address for a byte within the payload portion of the transmitted frame.

2. The communications interface as recited in claim 1, which may contain one or more scrambler/de-scramblers adapted to operate in byte-parallel mode.

5 3. The communications interface as recited in claim 1, wherein the first pointer control logic further comprises a table containing an offset to the start of one or more payloads within the received frame, and the second pointer control logic further comprises a table containing an offset to the start of one or more payloads within the transmitted frame.

10

4. The communications interface as recited in claim 3, wherein the table of the first control logic and the table of the second control logic each comprise a two-port random access memory.

15 5. The communications interface as recited in claim 1, wherein each queue manager further comprises a capture table containing a plurality of bits, such that each bit in the capture table of the first queue manager corresponds to a desired byte within the payload portion of the received frame, and each bit in the capture table of the second queue manager corresponds to a desired byte within the payload portion of the transmitted
20 frame.

6. The communications interface as recited in claim 1, wherein each of the first and second queue managers contains an overhead capture queue, such that the overhead capture queue of the first queue manager is adapted to transfer data from the computer
25 processor into the manifest portion of the transmitted frame, and the overhead capture queue of the second queue manager is adapted to remove data from the manifest portion of the received frame and transfer it to the computer processor.

7. The communications interface as recited in claim 2, wherein each of the first and second queue managers further comprises a plurality of data capture queues, for each of a plurality of lower-rate signals mapped into a payload within the transmitted and received frames.

5

8. The communications interface as recited in claim 1, wherein the transmitted and received frames comprise frames transmitted from or forwarded to a synchronous optical network (SONET) having differing transfer rates.

10 9. The communications interface as recited in claim 1, wherein the first pointer control logic further comprises first, second and third modulo counters, adapted to compute a frame number, row and column of each byte in the received data stream.

15 10. The communications interface as recited in claim 1, wherein the first and second queue managers each comprise a data port queue, such that each queue manager is adapted to respond to a set port instruction by configuring its data port queue according to parameters contained in the set port instruction.

20 11. The communications interface as recited in claim 10, wherein the first queue manager is adapted to respond to a broadcast bit in the set port instruction by copying, rather than removing, data from the received frame to the computer processor, and including said data in the transmitted frame.

25 12. The communications interface as recited in claim 1, wherein the interface is implemented as an integrated circuit.

13. A method for interfacing an optical communications network to a computer processor, comprising:

5 synchronously receiving a frame of serialized data from the network as an incoming data stream;

 removing at least some of the data from the received frame and transferring the removed data to the computer processor;

10 transferring data from the computer processor to a transmitted frame; and

 sending the transmitted frame as an outgoing data stream, such that the first byte of the outgoing data stream is adapted to be sent before the last byte of the incoming data stream has been received.

15

14. The method as recited in claim 13, further comprising queuing the incoming data stream in an input capture queue, and queuing the outgoing data stream in an output capture queue.

20 15. The method as recited in claim 13, further comprising scrambling the incoming data and de-scrambling the outgoing data, using a scrambler/de-scrambler circuit adapted to operate in byte-parallel mode.

25 16. The method as recited in claim 13, further comprising using first and second tables to record the offset to the start of one or more payload portions in the received and transmitted frames, respectively.

30 17. The method as recited in claim 13, further comprising using a frame index to record the physical address with respect to the received frame of a byte within the incoming serial data stream.

18. The method as recited in claim 13, further comprising using an index to record the logical address with respect to the payload portion of the received frame of a byte within the incoming byte data stream.

5 19. The method as recited in claim 16, further comprising implementing the first and second tables as two-port random access memory devices.

20. The method as recited in claim 14, further comprising associating a capture table with each capture queue, wherein the capture table comprises a plurality of bits, such that
10 each bit corresponds to a desired byte within the payload portion of the received and transmitted frames.

21. The method as recited in claim 13, further comprising using a first overhead capture queue adapted to transfer data from the manifest portion of the received frame to
15 the computer processor.

22. The method as recited in claim 13, further comprising using a second overhead capture queue adapted to transfer data from the computer processor to the manifest portion of the transmitted frame.

20 23. The method as recited in claim 14, further comprising using a plurality of data capture queues, for each of a plurality of lower-rate signals mapped into a payload within the received frame.

25 24. The method as recited in claim 14, further comprising using a plurality of data transmit queues, for each of a plurality of lower-rate signals mapped into a payload within the transmitted frame.

25. The method as recited in claim 13, wherein a frame comprises a data structure adapted for transmission over a synchronous optical network (SONET) having differing transfer rates.

5 26. The method as recited in claim 13, further comprising using first, second and third modulo counters adapted to compute the frame number, row and column of each byte in the incoming serial data stream.

27. The method as recited in claim 14, further comprising using a set port instruction
10 to configure the input capture queue and output capture queue.

28. The method as recited in claim 14, further comprising using a broadcast bit in the set port instruction to direct the input capture queue to copy, rather than remove, data from the received frame to the computer processor, and including said data in the
15 transmitted frame.

29. The method as recited in claim 13, further comprising using a changeable "fill byte" to replace data removed from the received frame, before transferring the data to the transmitted frame.
20